

## IN THE CLAIMS:

Please amend the claims as indicated below. The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

creating a graphical user interface for the graphical program in response to first user input;

displaying ~~an event structure~~ a first node for receiving user interface events in a block diagram for the graphical program in response to second user input; ~~and~~

receiving third user input specifying one or more user interface events to configure for the first node; and

configuring the ~~event structure~~ first node to receive ~~and respond to~~ the one or more user interface events specified by the third user input during execution of the graphical program.

2. (Currently Amended) The method of claim 1,

wherein the ~~event structure~~ first node comprises one or more sub-diagrams;

wherein ~~said configuring the event structure node to receive and respond to the one or more user interface events~~ the method further comprises configuring the one or more sub-diagrams to ~~receive and~~ respond to the one or more user interface events.

3. (Currently Amended) The method of claim 2,

wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input specifying one or more user interface events to which each of the sub-diagrams corresponds;

wherein, for each of the one or more sub-diagrams, said configuring the sub-diagram comprises:

~~specifying one or more user interface events to which the sub-diagram corresponds, in response to user input;~~

including graphical source code in the sub-diagram in response to user input, wherein the graphical source code is operable to respond to the one or more user interface events to which the sub-diagram corresponds.

4. (Original) The method of claim 3,  
wherein said including graphical source code in the sub-diagram comprises including two or more interconnected nodes in the sub-diagram.

5. (Original) The method of claim 1,  
wherein the block diagram comprises a data flow block diagram.

6. (Currently Amended) The method of claim 1, further comprising:  
executing the graphical program;  
wherein one or more user interface events to which the ~~event structure~~ first node is configured to receive ~~and respond~~ are generated during execution of the graphical program;

wherein the ~~event structure~~ first node is operable to receive ~~and respond to~~ the one or more user interface events generated during execution of the graphical program.

7. (Original) The method of claim 6,  
wherein the one or more user interface events generated during execution of the graphical program are generated in response to user input to the graphical user interface of the graphical program.

8. (Original) The method of claim 6,  
wherein, during execution of the graphical program, the block diagram executes on a first computer system and the graphical user interface is displayed on a display of a second computer system.

9. (Original) The method of claim 6,

wherein, during execution of the graphical program, the graphical user interface is displayed on a display of a computer system and the block diagram executes on a reconfigurable instrument connected to the computer system.

10. (Currently Amended) The method of claim 1,

wherein said configuring the ~~event-structure~~ first node to receive the one or more user interface events comprises configuring the ~~event-structure~~ first node to receive notification when the one or more user interface events are generated during execution of the graphical program.

11. (Currently Amended) The method of claim 1,

wherein said configuring the ~~event-structure~~ first node to receive the one or more user interface events comprises configuring the ~~event-structure~~ first node to receive information specifying occurrences of the one or more user interface events during execution of the graphical program.

12. (Currently Amended) The method of claim 1, further comprising:

~~wherein said configuring the event-structure node to receive and respond to the one or more user interface events comprises displaying graphical source code in the event structure~~ first node operable to ~~receive and~~ respond to the one or more user interface events.

13. (Currently Amended) The method of claim 1,

wherein said configuring the ~~event-structure~~ first node to receive ~~and respond to~~ the one or more user interface events comprises configuring the ~~event-structure~~ first node to receive ~~and respond to~~ a first user interface event;

wherein the first user interface event explicitly specifies a first user interface element of the graphical user interface and an action performed on the first user interface element.

14. (Original) The method of claim 13, wherein the first user interface element comprises one of:

- an indicator;
- a control;
- a menu element;
- a window.

15. (Currently Amended) The method of claim 1, further comprising:  
displaying a first graphical user interface for configuring the first node;  
wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input to the first graphical user interface to specify the one or more user interface events.

~~wherein said configuring the event structure node to receive and respond to one or more user interface events in response to user input comprises receiving user input via a graphical user interface dialog to specify the one or more user interface events.~~

16. (Currently Amended) The method of claim 1, further comprising:  
displaying an event registration a second node for dynamically registering user interface events in the block diagram in response to user input;

wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input specifying a first user interface event to dynamically register during execution of the graphical program;

wherein the method further comprises configuring the ~~event registration~~ second node to dynamically register a the first user interface event during execution of the graphical program;

wherein, before said dynamically registering the first user interface event, the first node is not operable to receive the first user interface event;

wherein, after said dynamically registering the first user interface event, the ~~event structure~~ first node is operable to receive ~~and respond to~~ the first user interface event.

17. (Currently Amended) The method of claim 16,

wherein said configuring the ~~event-registration~~ second node to dynamically register a the first user interface event during execution of the graphical program comprises connecting the ~~event-registration~~ second node to the ~~event-structure~~ first node in response to user input.

18. (Currently Amended) The method of claim 1, ~~further comprising:~~

wherein the one or more user interface events specified by the third user input includes a first user interface event;

wherein the method further comprises displaying an ~~event-un-registration~~ a second node for dynamically un-registering user interface events in the block diagram in response to user input;

configuring the ~~event-un-registration~~ second node to dynamically un-register a the first user interface event during execution of the graphical program;

wherein, before said dynamically un-registering the first user interface event, the first node is operable to receive the first user interface event;

wherein, after said dynamically un-registering the first user interface event, the ~~event-structure~~ first node ~~does~~ is not operable to receive ~~and respond to~~ the first user interface event.

19. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

displaying ~~an event-structure~~ a first node for receiving programmatic events in a block diagram for the graphical program in response to first user input; ~~and~~

receiving second user input specifying one or more programmatic events to configure for the first node; and

configuring the ~~event-structure~~ first node to receive ~~and respond to~~ the one or more programmatic events specified by the second user input during execution of the graphical program.

20. (Currently Amended) The method of claim 19,  
wherein the ~~event structure~~ first node comprises one or more sub-diagrams;  
wherein said ~~configuring the event structure node to receive and respond to the~~  
~~one or more programmatic events~~ the method further comprises configuring the one or  
more sub-diagrams to ~~receive and~~ respond to the one or more programmatic events.

21. (Currently Amended) The method of claim 20,  
wherein said receiving the second user input specifying the one or more  
programmatic events to configure for the first node comprises receiving user input  
specifying one or more programmatic events to which each of the sub-diagrams  
corresponds;

wherein, for each of the one or more sub-diagrams, said configuring the sub-  
diagram comprises:

~~specifying one or more programmatic events to which the sub-diagram~~  
~~corresponds, in response to user input;~~

including graphical source code in the sub-diagram in response to user input,  
wherein the graphical source code is operable to respond to the one or more  
programmatic events to which the sub-diagram corresponds.

22. (Original) The method of claim 19, wherein the one or more programmatic  
events comprise one or more of:

a user interface event;

a system event;

a timer event;

an event generated in response to data acquired from a device.

23. (Currently Amended) A computer-implemented method for creating a  
graphical program, the method comprising:

creating a graphical user interface for the graphical program in response to user  
input;

creating a block diagram for the graphical program in response to user input; ~~and~~  
receiving user input specifying one or more user interface events; and  
configuring the graphical program to receive ~~and respond to the~~ one or more user  
interface events in response to the user input specifying the one or more user interface  
events.

24. (Original) The method of claim 23,  
wherein said creating the block diagram for the graphical program in response to  
user input comprises arranging a plurality of nodes on a display and interconnecting the  
plurality of nodes in response to user input.

25. (Currently Amended) The method of claim 23, further comprising:  
executing the graphical program;  
wherein one or more user interface events to which the graphical program is  
configured to receive ~~and respond~~ are generated during execution of the graphical  
program;  
wherein the graphical program is operable to receive ~~and respond to~~ the one or  
more user interface events generated during execution of the graphical program.

26. (Currently Amended) The method of claim 23,  
wherein said configuring the graphical program to receive ~~and respond to the~~ one  
or more user interface events comprises configuring the block diagram to receive ~~and~~  
~~respond to~~ the one or more user interface events.

27. (Currently Amended) The method of claim 23, further comprising:  
~~wherein said configuring the block diagram to receive and respond to one or more~~  
~~user interface events comprises~~ including graphical source code in the block diagram  
operable to ~~receive and~~ respond to the one or more user interface events.

28. (Currently Amended) The method of claim 23, further comprising:

displaying a first graphical user interface for selecting user interface events,  
wherein said receiving the user input specifying the one or more user interface events  
comprises receiving user input to the first graphical user interface to select the one or  
more user interface events;

wherein said configuring the graphical program to receive ~~and respond to~~ the one  
or more user interface events ~~in response to user input comprises receiving user input via~~  
a is performed in response to said receiving user input to the first graphical user interface  
dialog to specify select the one or more user interface events.

29. (Currently Amended) The method of claim 24 23, further comprising:

including a first node in the block diagram of the graphical program;

wherein said configuring the graphical program to receive ~~and respond to~~ the one  
or more user interface events comprises configuring the first node to receive the one or  
more user interface events in response to the user input specifying the one or more user  
interface events ~~comprises including an event structure node in the block diagram in~~  
~~response to user input;~~

~~wherein the event structure node is operable to receive and respond to the one or~~  
~~more user interface events.~~

30. (Currently Amended) The method of claim 29,

wherein the ~~event structure~~ first node includes one or more sub-diagrams;

wherein each sub-diagram includes graphical source code specifying a response to  
one or more user interface events.

31. (Canceled)

32. (Currently Amended) A memory medium for creating a graphical program,  
the memory medium comprising program instructions executable to:

create a graphical user interface for the graphical program in response to first user  
input;



display ~~an event structure~~ a first node for receiving user interface events in a block diagram for the graphical program in response to second user input; ~~and~~  
receive third user input specifying one or more user interface events to configure for the first node; and  
configure the ~~event structure~~ first node to receive ~~and respond to~~ the one or more user interface events specified by the third user input during execution of the graphical program.

33. (Currently Amended) The memory medium of claim 32,  
wherein the ~~event structure~~ first node comprises one or more sub-diagrams;  
wherein ~~said configuring the event structure node to receive and respond to the one or more user interface events~~ comprises the program instructions are further executable to configure ~~configuring~~ the one or more sub-diagrams to ~~receive and respond to the one or more user interface events.~~

34. (Currently Amended) The memory medium of claim 33,  
wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input specifying one or more user interface events to which each of the sub-diagrams corresponds;  
wherein, for each of the one or more sub-diagrams, said configuring the sub-diagram comprises:  
~~specifying one or more user interface events to which the sub-diagram corresponds, in response to user input;~~  
including graphical source code in the sub-diagram in response to user input, wherein the graphical source code is operable to respond to the one or more user interface events to which the sub-diagram corresponds.

35. (Original) The memory medium of claim 34,  
wherein said including graphical source code in the sub-diagram comprises including two or more interconnected nodes in the sub-diagram.

36. (Currently Amended) A computer-implemented method for creating a graphical program, the method comprising:

displaying a first node in a block diagram of the graphical program in response to user input; ~~and~~

associating graphical source code with the first node in response to user input;

associating a first user interface event with the first node in response to user input;

and

configuring the graphical source code associated with the first node to execute in response to the first user interface event associated with the first node.

~~wherein the graphical source code is operable to execute in response to a user interface event.~~

37. (Currently Amended) The method of claim 36,

wherein said graphical source code executing in response to the first user interface event comprises the graphical source code executing in response to the first user interface event during execution of the graphical program, wherein the first user interface event is generated during execution of the graphical program.

38. (Previously Presented) The method of claim 36,

wherein the graphical source code comprises a plurality of interconnected nodes.

39. (Previously Presented) The method of claim 38,

wherein said associating the graphical source code with the first node comprises:

displaying the plurality of nodes in response to user input; and

interconnecting the plurality of nodes in response to user input.

40. (Currently Amended) The method of claim 36,

wherein said associating the graphical source code with the first node in response to user input comprises displaying the graphical source code within the first node in response to user input, wherein the graphical source code associated with the first node is visible in the block diagram of the graphical program.

41. (Currently Amended) The method of claim 36, ~~further comprising:~~  
wherein said associating the first user interface event with the first node  
comprises receiving user input specifying one or more the first user interface events event  
to associated associate with which the first node corresponds;  
~~wherein the graphical source code is operable to respond to the one or more user~~  
~~interface events to which the first node corresponds.~~

42. (Currently Amended) The method of claim ~~36~~ 41,  
wherein said associating graphical source code with the first node in response to  
user input comprises associating two or more portions of graphical source code with the  
first node in response to user input;  
wherein said configuring the graphical source code associated with the first node  
to execute in response to the first user interface event comprises configuring a first  
portion from the two or more portions of graphical source code to execute in response to  
the first user interface event;  
wherein the method further comprises:  
associating a second user interface event with the first node in response to user  
input; and  
configuring a second portion from the two or more portions of graphical source  
code to execute in response to the second user interface event.  
~~wherein each of the portions of graphical source code is operable to respond to~~  
~~one or more of the one or more user interface events.~~

43. (Currently Amended) The method of claim 41,  
wherein said receiving user input specifying the first ~~one or more~~ user interface  
~~events event to which~~ associate with the first node ~~corresponds~~ comprises receiving user  
input specifying a name of ~~each of the~~ first user interface ~~events event~~.

44. (Currently Amended) The method of claim 41, further comprising:  
displaying a graphical user interface dialog;

wherein said receiving user input specifying the first ~~one or more~~ user interface events event to which associate with the first node ~~corresponds~~ comprises receiving user input via the graphical user interface dialog to specify the first ~~one or more~~ user interface events event.

45. (Canceled)

46. (Currently Amended) The method of claim 36,  
wherein the graphical program has a graphical user interface;  
wherein the first user interface event is associated with a first user interface element of the graphical user interface.

47. (Previously Presented) The method of claim 46, wherein the first user interface element comprises one of:

- an indicator;
- a control;
- a menu element;
- a window.

48. (Currently Amended) The method of claim 36,  
wherein the graphical program includes a graphical user interface;  
wherein the first user interface event is associated with a user action performed on the graphical user interface.

49. (Currently Amended) The method of claim 36, further comprising:  
executing the graphical program;  
generating the first user interface event during execution of the graphical program;

wherein said executing the graphical program includes executing the graphical source code associated with the first node in response to said generating the first user interface event.

50. (Currently Amended) The method of claim 36,  
wherein the graphical program includes a graphical user interface;  
wherein the method further comprises:

executing the graphical program;

generating the first user interface event during execution of the graphical program, wherein said generating the first user interface event comprises generating the first user interface event in response to user input to the graphical user interface;

wherein said executing the graphical program includes executing the graphical source code associated with the first node in response to said generating the first user interface event.

51. (Previously Presented) The method of claim 36,  
wherein the block diagram of the graphical program comprises a data flow diagram.

52. (Previously Presented) The method of claim 36,  
wherein the block diagram comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

53. (Currently Amended) A memory medium for creating a graphical program,  
the memory medium comprising program instructions executable to:

display a first node in a block diagram of the graphical program in response to user input; ~~and~~

associate graphical source code with the first node in response to user input;

associate a first user interface event with the first node in response to user input;

and

configure the graphical source code associated with the first node to execute in response to the first user interface event associated with the first node.

~~wherein the graphical source code is operable to execute in response to a user interface event.~~

54. (Currently Amended) The memory medium of claim 53,  
wherein said graphical source code executing in response to the first user interface event comprises the graphical source code executing in response to the first user interface event during execution of the graphical program, wherein the first user interface event is generated during execution of the graphical program.

55. (Previously Presented) The memory medium of claim 53,  
wherein the graphical source code comprises a plurality of interconnected nodes.

56. (Previously Presented) The memory medium of claim 55,  
wherein said associating the graphical source code with the first node comprises:  
displaying the plurality of nodes in response to user input; and  
interconnecting the plurality of nodes in response to user input.

57. (Currently Amended) The memory medium of claim 53,  
wherein said associating the graphical source code with the first node in response to user input comprises displaying the graphical source code within the first node in response to user input, wherein the graphical source code associated with the first node is visible in the block diagram of the graphical program.

58. (Currently Amended) The memory medium of claim 53,  
wherein said associating the first user interface event with the first node ~~the memory medium further comprises program instructions executable to receive~~ receiving user input specifying ~~one or more~~ the first user interface events event to associate with which the first node corresponds;  
~~wherein the graphical source code is operable to respond to the one or more user interface events to which the first node corresponds.~~

59. (Currently Amended) The memory medium of claim 53 ~~58~~,

wherein said associating graphical source code with the first node in response to user input comprises associating two or more portions of graphical source code with the first node in response to user input;

wherein said configuring the graphical source code associated with the first node to execute in response to the first user interface event comprises configuring a first portion from the two or more portions of graphical source code to execute in response to the first user interface event;

wherein the program instructions are further executable to:

associate a second user interface event with the first node in response to user input; and

configure a second portion from the two or more portions of graphical source code to execute in response to the second user interface event.

~~wherein each of the portions of graphical source code is operable to respond to one or more of the one or more user interface events.~~

60. (Currently Amended) The ~~method~~ memory medium of claim 58,

wherein the memory medium further comprises program instructions executable to display a graphical user interface dialog;

wherein said receiving user input specifying the ~~one or more~~ first user interface ~~events~~ event to ~~which~~ associate with the first node ~~corresponds~~ comprises receiving user input via the graphical user interface dialog to specify the first ~~one or more~~ user interface ~~events~~ event.

61. (Canceled)

62. (Currently Amended) The memory medium of claim 53,

wherein the graphical program includes a graphical user interface;

wherein the first user interface event is associated with a first user interface element of the graphical user interface.

63. (Currently Amended) The memory medium of claim 53,  
wherein the graphical program includes a graphical user interface;  
wherein the first user interface event is associated with a user action performed on  
the graphical user interface.

64. (Previously Presented) The memory medium of claim 53,  
wherein the block diagram of the graphical program comprises a data flow  
diagram.

65. (Previously Presented) The memory medium of claim 53,  
wherein the block diagram comprises a plurality of interconnected nodes that  
visually indicate functionality of the graphical program.

66. (Currently Amended) A system for creating a graphical program, the system  
comprising:

a memory storing program instructions;

a processor coupled to the memory; and

a display device;

wherein the processor is operable to execute the program instructions stored in the  
memory to:

display a first node on the display device in response to user input,  
wherein said displaying the first node comprises displaying the first node in a block  
diagram of the graphical program; ~~and~~

associate graphical source code with the first node in response to user  
input;

associate a first user interface event with the first node in response to user  
input; and

configure the graphical source code associated with the first node to  
execute in response to the first user interface event associated with the first node.

~~wherein the graphical source code is operable to execute in response to a user~~  
~~interface event.~~



67. (New) The method of claim 1, further comprising:

displaying a list of user interface events;

wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input to select the one or more user interface events from the displayed list of user interface events.

68. (New) The method of claim 1,

wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input specifying names of the one or more user interface events to configure for the first node.

69. (New) The method of claim 1,

wherein the first node comprises a plurality of sub-diagrams;

wherein the method further comprises configuring each of the sub-diagrams to respond to one or more of the one or more user interface events specified by the third user input.

70. (New) The method of claim 1, further comprising:

displaying graphical source code in the graphical program operable to respond to the one or more user interface events.

71. (New) The method of claim 1,

wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input specifying a first user interface event to be dynamically registered during execution of the graphical program;

wherein the method further comprises configuring the graphical program to dynamically register the first user interface event during execution of the graphical program;

wherein, before said dynamically registering the first user interface event during execution of the graphical program, the first node is not operable to receive the first user interface event;

wherein said dynamically registering the first user interface event enables the first node to receive the first user interface event.